

Куліковська О. В.

студентка фізико-математичного факультету

Т.А. Вакалюк

канд. пед. наук,

доцент кафедри прикладної математики та інформатики

Житомирського державного університету імені Івана Франка

MYSQL ЧИ SQLITE ВИКОРИСТОВУВАТИ НА БАЗІ DJANGO CMS

Анотація: *розглянуто особливості створення сайтів на базі Django CMS. Досліджено використання баз даних MySQL та SQLite в даній CMS.*

Ключові слова: *Django CMS, MySQL, SQLite, база даних.*

Аннотация: рассмотрены особенности создания сайтов на базе Django CMS. Исследовано использование баз данных MySQL и SQLite в данной CMS.

Ключевые слова: Django CMS, MySQL, SQLite, база данных.

Summary: There are was considered features of a site based on Django CMS. We explored the use of databases MySQL and SQLite in Django CMS.

Key words: Django CMS, MySQL, SQLite, database.

Зі збільшенням потреб створюються нові способи вирішення завдань. Широкий спектр можливостей сфери веб-технологій відкривається у результаті впровадження систем керування вмістом (Content Management System, CMS).

Використання CMS полегшує розробку серверної частини (бекенду). Django CMS є однією з платформ систем управління контентом з вільним і відкритим вихідним кодом. Вона заснована на Джанго і має наступні особливості:

- надійна підтримка інтернаціоналізації (i18n) для створення багатомовних сайтів;
- редагування фронтенду, що забезпечує швидкий доступ до інтерфейсу управління контентом;
- підтримка різних редакторів з розширеними можливостями для редагування тексту;
- гнучка система плагінів, що дозволяє розробникам ставити потужні інструменти в руках редакторів, не перевантажуючи їх з важким інтерфейсом;
- підтримка PostgreSQL, PostgreSQL, MySQL, SQLite, Oracle баз даних.

Нова версія Django 1.9 має деякі доповнення:

- підтримка виконання дій (сигналів) після коммітов транзакції;
- підтримка валідації паролів;
- нові стилі для адмінки;
- підтримка паралельного виконання тестів [1].

Запустити сайт на базі Django можна і без бази даних. Якщо ви просто хочете обробляти динамічні сторінки, які не звертаються до бази даних, все це можливо. Але в реальному проекті без неї не обійтися.

Слід взяти до уваги, що деякі додаткові інструменти поставляються з Django вимагають наявності бази даних. Відповідно, якщо ви відмовляєтеся від використання бази даних, ви також відмовляєтеся від використання цих інструментів [3].

Використання Django з MySQL

Django підтримує MySQL 5.5 і вище.

Функція Django `inspectdb` використовує базу даних `INFORMATION_SCHEMA`, що містить ґрунтовні дані про всі схеми баз даних.

Django передбачає, що база даних підтримує Unicode (UTF-8 кодування) і покладає на нього завдання забезпечення дотримання транзакцій і цілісності.

MySQL має декілька підсистем зберігання даних (двигунів). Ви можете змінити двигун зберігання за замовчуванням в конфігурації сервера.

В MySQL 5.5.5 підсистема зберігання даних зберігання за замовчуванням є InnoDB. Цей двигун повністю підтримує транзакційні і зовнішні ключі посилення. Це, ймовірно, найкращий вибір в цій точці. Проте, варто зауважити, що лічильник автоінкремент InnoDB втрачається при перезапуску MySQL, тому що він не пам'ятає значення `AUTO_INCREMENT`, замість того, щоб відтворити його як `"max (ID) + 1"`. Це може привести до ненавмисного повторного використання значень `AutoField`.

При оновленні існуючого проекту MySQL 5.5.5, після чого додати кілька таблиць, необхідно переконатися, що таблиці використовують один і той же механізм зберігання (тобто MyISAM проти InnoDB). Зокрема, якщо таблиці, які мають `ForeignKey` між ними використовують різні механізми зберігання, система видасть помилку [5].

Використання Django з SQLite

SQLite 3 забезпечує відмінну альтернативу розробки для додатків, які в основному ризначені тільки для читання або необхідна найменша установка.

Робота як і з усіма серверами баз даних, однак є деякі відмінності, які є специфічними для SQLite.

Для всіх версій SQLite прослідковується трохи нелогічна поведінка при зіставленні деяких типів рядків. Вони спрацьовують при використанні іехаст або містять фільтри в Querysets. Поведінка розбивається на два випадки:

1. Для відповідності підрядка, всі пари стають незалежні від регістру. Тобто фільтр (`name__contains = "aa"`) буде відповідати назву "Aabb".

2. Для рядків, що містять символи поза діапазону ASCII, всі точні рядкові пари проводиться з урахуванням регістру, навіть коли без урахування регістру параметри будуть передаватися в запит. Таким чином, іехаст фільтр буде вести себе точно так само, як точний фільтр в цих випадках.

Існують деякі шляхи, що дозволяють обійти дані незручності, описані в sqlite.org, але вони не використовуються за замовчуванням SQLite бекенда в Django. Таким чином, Django використовує параметри SQLite за замовчуванням, тому при роботі варто брати до уваги чутливість до регістру та фільтрацію [5].

Важливу перевагу відіграє налаштування бази даних SQLite, оскільки з нею не виникає труднощів. Сама база при цьому зберігається в одному файлі на диску. SQLite не варто використовувати в робочих сайтах з реальним навантаженням, але цілком можна для тестування і розробки функціоналу.

Щоб використовувати `sqlite`, досить вказати `django.db.backends.sqlite3` як ENGINE і прописати ім'я файлу, в якому буде зберігатися база, в параметрі NAME [4].

Отже, серед двох розглянутих баз даних SQLite та MySQL я надаю перевагу MySQL при створенні великих проектів. MySQL дуже функціональна, вільно розповсюджується, успішно працює з різними сайтами і веб додатками.

Використання SQLite є особливим випадком: не потрібно створювати базу даних, так як SQLite використовує файли на файлової системи для зберігання своїх даних, що вигідно на простих або невеликих проектах.

MySQL пропонує досить багато інструментів для розробки додатків. Так як це серверна СУБД, додатки для доступу до даних, на відміну від SQLite працюють зі службами MySQL.

Використані джерела:

1. Релиз Django 1.9. [Електронний ресурс]. – Режим доступу: <https://habrahabr.ru/post/272137/>
2. Docs. Development & community. Contributing code. [Електронний ресурс]. – Режим доступу: <http://docs.djangoproject.com/en/latest/contributing/code.html>
3. Настройка базы данных. [Електронний ресурс]. – Режим доступу: <http://djbook.ru/ch02s04.html>
4. Начинаем работу с Django [Електронний ресурс]. – Режим доступу: <http://itman.in/django-howto/#i-4>
5. Django documentation. Databases. [Електронний ресурс]. – Режим доступу: <https://docs.djangoproject.com/ja/1.9/ref/databases/>